Amendments to the Claims:

The claims below replace all prior versions and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method of establishing an interface for a user system between a remote service that does not have a user interface and ~~an~~ a remote application comprising:

receiving a file by the application from a the user system, wherein the file contains standardized interface data;

~~providing~~ sending the file from the application to the service;

generating a return file by the service, wherein the return file contains standardized interface data;

~~providing~~ sending the return file to the application with a dynamic user interface specification containing explanatory error messages regarding graphics in the file; ~~and~~

using extensible style language transformation (XSLT) to convert any multi-part Multipurpose Internet Mail Extensions (MIME) encoding of the dynamic user interface into extensible mark-up language (XML) encoding;

using the dynamic user interface specification to generate a graphical user interface by the application; and

creating communication between the service that does not have a graphical user interface and the user by providing the return file and the explanatory error messages in the graphical user interface to the user system.

2. (original) The method of establishing an interface between a service and an application of claim 1 wherein the return file is presented as a browser interface.

3. (original) The method of establishing an interface between a service and an application of claim 1 further comprising:

generating a dynamic user interface specification by the service;

providing the dynamic user interface specification to application;

generating a user interface response by the application; and

providing the user interface response to the service.

4. (original) The method of establishing an interface between a service and an application of claim 3 wherein the return file is presented as a browser interface.

5. (original) The method of establishing an interface between a service and an application of claim 3 wherein the user system determines content of the user interface response.

6. (original) The method of establishing an interface between a service and an application of claim 5 wherein the return file is presented as a browser interface.

7. (original) The method of establishing an interface between a service and application of claim 3 wherein the user interface specification and user interface response are written in a markup language.

8. (original) The method of establishing an interface between a service and application of claim 4 wherein the user interface specification and user interface response are written in a markup language.

9. (original) The method of establishing an interface between a service and application of claim 5 wherein the user interface specification and user interface response are written in a markup language.

10. (original) The method of establishing an interface between a service and application of claim 6 wherein the user interface specification and user interface response are written in a markup language.

11. (currently amended) A system for establishing an interface comprising of:

a user system;

an application that receives a file the user system, wherein the file contains standardized interface data; and

a service that does not have a user interface configured to receives the file and generates a return file containing standardized interface data, sending the return file to the application and the user system;

wherein the return file contains a dynamic user interface specification with explanatory error messages regarding graphics in the file, extensible style language transformation (XSLT) is used to convert any multi-part Multipurpose Internet Mail Extensions (MIME) encoding of the dynamic user interface into extensible mark-up language (XML) encoding, and the application is configured to use the dynamic user interface specification to generate a graphical user interface for creating communication between the service that does not have a graphical user interface and the user by providing the return file and the explanatory error messages in the graphical user interface to the user system.

12. (original) The system for establishing an interface of claim 11 wherein the return file is presented as a browser interface.

13. (original) The system for establishing an interface of claim 11 further comprised of:

a dynamic user interface specification generated by the service, wherein the dynamic user interface specification is provided to the application; and

a user interface response generated by the application; wherein the user interface response is provided to the service.

14. (original) The system for establishing an interface of claim of claim 13 wherein the return file is presented as a browser interface.

15. (original) The system for establishing an interface of claim of claim 13 wherein the user system determines content of the user interface response.

16. (original) The system for establishing an interface of claim of claim 15 wherein the return file is presented as a browser interface.

17. (original) The system for establishing an interface of claim of claim 13 wherein the user interface specification and user interface response are written in a markup language.

18. (original) The system for establishing an interface of claim of claim 14 wherein the user interface specification and user interface response are written in a markup language.

19. (original) The system for establishing an interface of claim of claim 15 wherein the user interface specification and user interface response are written in a markup language.

20. (original) The system for establishing an interface of claim of claim 16 wherein the user interface specification and user interface response are written in a markup language.

21. (currently amended) A computer system comprising:

a processor; a computer;

computer readable medium coupled to the processor; and

computer code encoded in the computer readable medium, configured to cause the processor to:

receive a file by the application from a user system, wherein the file contains standardized interface data;

provide the file to ~~the~~ a service that does not have a graphical user interface;

generate a return file by the service, wherein the return file contains standardized interface data;

provide the return file to the application; and

provide the return file to the user system;

wherein the return file contains a dynamic user interface specification with explanatory error messages regarding graphics in the file, extensible style language transformation (XSLT) is used to convert any multi-part Multipurpose Internet Mail Extensions (MIME) encoding of the dynamic user interface into extensible mark-up language (XML) encoding, and the application is configured to use the dynamic UI specification to generate a graphical user interface for creating communication between the service that does not have a graphical user interface and the user by providing the return file and the explanatory error messages in the graphical user interface to the user system.

22. (original) The computer system of claim 21 wherein the return file is presented as a browser interface.

23. (original) The computer system of claim 21 wherein the processor further:

generates a dynamic user interface specification by the service;

provides the dynamic user interface specification to application;

generates a user interface response by the application; and

provides the user interface response to the service.

24. (original) The computer system of claim 20 wherein the configuration file is written in an extensible markup language.

25. (original) The computer system of claim 23 wherein the user system determines content of the user interface response.

26. (original) The computer system of 25 wherein the return file is presented as a browser interface.

27. (original) The computer system of claim 23 wherein the user interface specification and user interface response are written in a markup language.

28. (original) The computer system of claim 24 wherein the user interface specification and user interface response are written in a markup language.

29. (original) The computer system of claim 25 wherein the user interface specification and user interface response are written in a markup language.

30. (original) The computer system of claim 26 wherein the user interface specification and user interface response are written in a markup language.

31. (currently amended) An apparatus for establishing an interface comprising:

means for receiving a file by the application from a user system, wherein the file contains standardized interface data;

means for providing the file to ~~the~~ a service that does not have a graphical user interface;

means for generating a return file by the service, wherein the return file contains standardized interface data;

means for providing the return file to the application; ~~and~~

means for providing the return file to the user system;

means for creating a dynamic user interface specification in the return file with explanatory error messages regarding graphics in the file;

means for using the dynamic user interface specification to generate a graphical user interface;

means for using extensible style language transformation (XSLT) to convert any multi-part Multipurpose Internet Mail Extensions (MIME) encoding of the dynamic user interface into extensible mark-up language (XML) encoding; and

means for creating communication between the service that does not have a graphical user interface and the user by providing the return file and the explanatory error messages in the graphical user interface to the user system.

32. (original) The apparatus of claim 31 wherein the return file is presented as a browser interface.

33. (original) The apparatus of claim 31 further comprising:

means for generating a dynamic user interface specification by the service;

means for providing the dynamic user interface specification to application;

means for generating a user interface response by the application; and

means for providing the user interface response to the service.

34. (original) The apparatus of claim 33 wherein the return file is presented as a browser interface.

35. (original) The apparatus of claim 33 wherein the user system determines content of the user interface response.

36. (original) The apparatus of claim 35 wherein the return file is presented as a browser interface.

37. (original) The apparatus of claim 33 wherein the user interface specification and user interface response are written in a markup language.

38. (original) The apparatus of claim 34 wherein the user interface specification and user interface response are written in a markup language.

39. (original) The apparatus of claim 35 wherein the user interface specification and user interface response are written in a markup language.

40. (original) The apparatus of claim 36 wherein the user interface specification

and user interface response are written in a markup language.

41. (currently amended) A computer program product encoded in computer readable media, the computer program product comprising:

a first set of instructions, executable on a computer system, configured to receive a file by the application from a user system, wherein the file contains standardized interface data;

a second set of instructions, executable on a computer system, configured to provide the file to ~~the~~ a service that does not have a graphical user interface;

a third set of instructions, executable on a computer system, configured to generate a return file by the service, wherein the return file contains standardized interface data;

a fourth set of instructions, executable on a computer system, configured to provide the return file to the application; and

a fifth set of instructions, executable on a computer system, configured to provide the return file to the user system;

wherein the return file contains a dynamic user interface specification with explanatory error messages regarding graphics in the file, using extensible style language transformation (XSLT) to convert any multi-part Multipurpose Internet Mail Extensions (MIME) encoding of the dynamic user interface into extensible mark-up language (XML) encoding, and the application is configured to use the dynamic user interface specification to generate a graphical user interface for creating communication between the service that does not have a graphical user interface and the user by providing the return file and the explanatory error messages in the graphical user interface to the user system.

42. (original) The computer program product of claim 41 wherein the return file is presented as a browser interface.

43. (original) The computer program product of claim 41 further comprising:

a fifth set of instructions, executable on a computer system, configured to

generate a dynamic user interface specification by the service;

a sixth set of instructions, executable on a computer system, configure to provide the dynamic user interface specification to application;

a seventh set of instructions, executable on a computer system, configure to generate a user interface response by the application; and

an eighth set of instructions, executable on a computer system, configure to provide the user interface response to the service.

44. (original) The computer program product of claim 40 wherein the configuration file is written in an extensible markup language.

45. (original) The computer program product of claim 43 wherein the user system determines content of the user interface response.

46. (original) The computer program product of claim 45 wherein the return file is presented as a browser interface.

47. (original) The computer program product of claim 43 wherein the user interface specification and user interface response are written in a markup language.

48. (original) The computer program product of claim 44 wherein the user interface specification and user interface response are written in a markup language.

49. (original) The computer program product of claim 45 wherein the user interface specification and user interface response are written in a markup language.

50. (original) The computer program product of claim 46 wherein the user interface specification and user interface response are written in a markup language.